Weekly Report(March.19,2019-April.17,2019)

Rui Shaopu

Abstract

During these weeks, I finished assignment3. I learned RNN, LSTM and some other interesting models. And I also modified the last part of Pytorch.ipnb in assignment2.

1 Learned

After finishing all the thing about them, I have to say, it's easy to finish code if you know the formula. But it'e hard to understand how they work and why they can make it.

1.1 RNN

1.1.1 Introduction

Recurrent Neural Network. It's a typical model to address squential data, such as videos, sentences, music and so on.

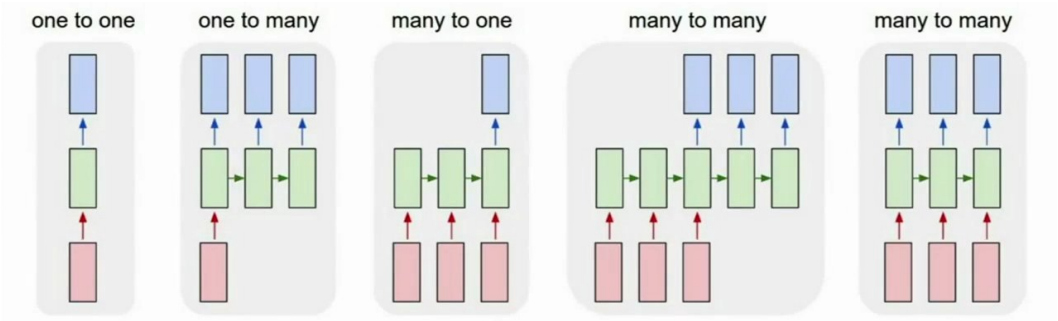

Figure 1: Different kinds of RNN

Formula:

for each hidden layer:

$$h_t = f_w(h_{t-1}, x_t) = tanh(Wh_{t-1} * h_{t-1} + Wx * X + b)$$
$$y_t = W_{hy}h_t$$

$h_t$is the new state, while $h_{t-1}$is the old state, and x_t is input of this timestep. f_w is some function with parameters W, the right part of the equation is the function used in homework. y_t will be used when calculating loss.

1.1.2 My work

Actually It's much more easier than other parts, so I will just show the result.

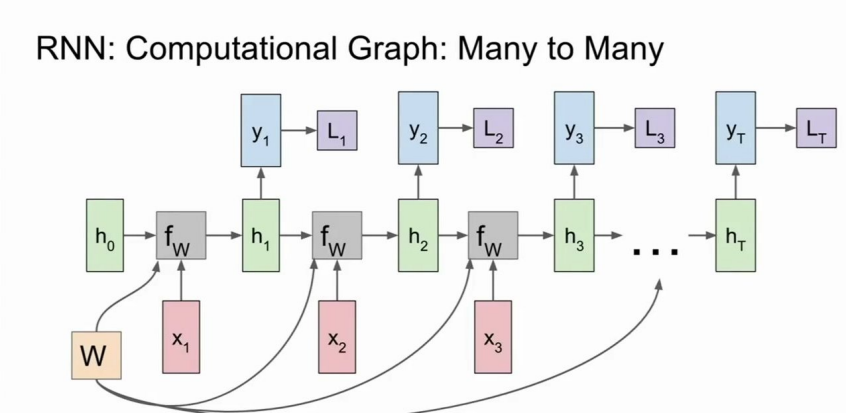

Figure 2: Typical many-to-many model

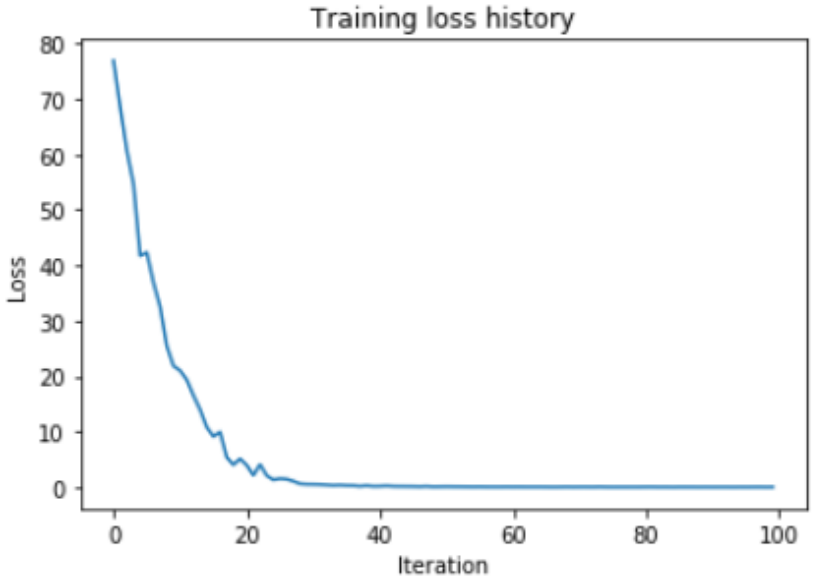

Figure 3: loss

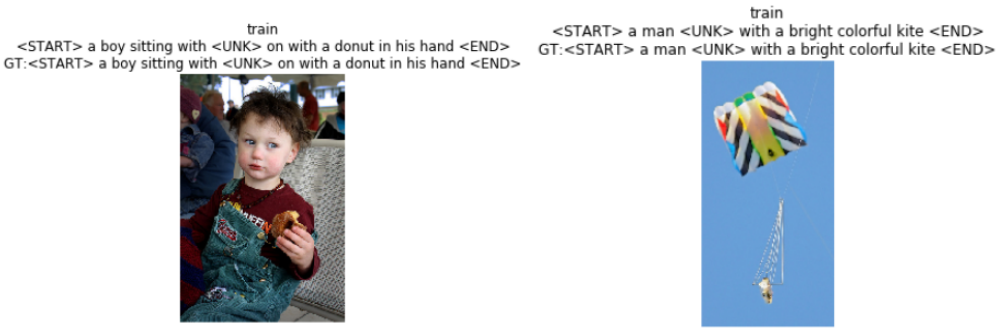

Figure 4: trained result

1.2 LSTM

1.2.1 Introduction

Long Short Term Memory, is slightly fancier recurrence relationfor RNN. It's designed to solve the problem of vanishing and exploding gradients.

Formula:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot tanh(c_t)$$

In this formulas, i means *input_gate*, f means *forget_gate*, o means *output_gate*, while g doesn't have a good name.

1.2.2 My work

The implementation is the same as RNN, so I will still say nothing about it.

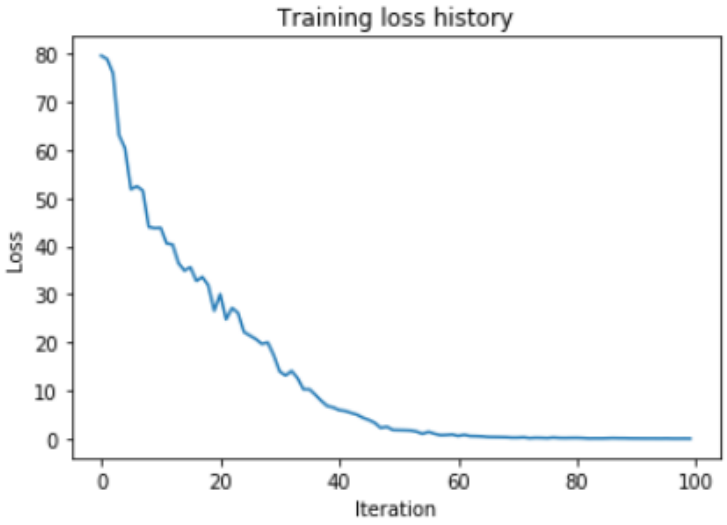

Figure 5: Final result

Figure 6: trained results

2 Modification

After finishing assignment3, I think back "open-ended challenge" in assignment2. Since it really attracted me and I think I can find a better model to get better grade.

I try to modify the model for several trials, I finally get about 80% accuracy. But I have some questions showed below.

```
Iteration 300, loss = 0.0170
Checking accuracy on validation set
Got 803 / 1000 correct (80.30)

Iteration 400, loss = 0.0037
Checking accuracy on validation set
Got 796 / 1000 correct (79.60)

Iteration 500, loss = 0.0584
Checking accuracy on validation set
Got 806 / 1000 correct (80.60)

Iteration 600, loss = 0.0446
Checking accuracy on validation set
Got 788 / 1000 correct (78.80)

Iteration 700, loss = 0.0138
Checking accuracy on validation set
Got 788 / 1000 correct (78.80)
```

Figure 7: open-ended challenge accuracy

3 Understanding

RNN addresses sequential words in the homework. In order to predict next layer's state, it will use all of the previous state as input. It works very well. However, it has the problem of vanishing and exploding gradients. That's why LSTM exists. LSTM will forget part of previous hidden state, which makes the network work better.

Question: As training goes on, I come to confuse about the relation between an actual layer and its functions. For example, why adding convolutional layers in a CNN will get better performance? I know there must be some scientific basis, so how to use this basis if I want to improve a model?

4 Plan

I need to focus on my professional courses.